USL / DBMS NASA / RECON

WORKING PAPER SERIES

Report Number

DBMS.NASA/RECON-22

The USL/DBMS NASA/RECON Working Paper Series contains a collection of reports representing results of activities being conducted by the Center for Advanced Computer Studies of the University of Southwestern Louisiana pursuant to the specifications of National Aeronautics and Space Administration Contract Number NASW-3846. The work on this contract is being performed jointly by the University of Southwestern Louisiana and Southern University.

For more information, contact:

Wayne D. Dominick

Editor
USL/DBMS NASA/RECON Working Paper Series
Center for Advanced Computer Studies
University of Southwestern Louisiana
P. O. Box 44330
Lafayette, Louisiana 70504

(NASA-CR-184530) KARL: A KNOWLEDGE-ASSISTED RETRIEVAL LANGUAGE. PRESENTATION VISUALS M.S. Thesis Final Report, 1 Jul. 1985 - 31 Dec. 1987 (University of Southwestern Louisiana. Lafayette. Center for Advanced   N89-14570

Unclas
G3/82 0183568

KARL:
A KNOWLEDGE-ASSISTED RETRIEVAL LANGUAGE

PRESENTATION VISUALS

A Thesis
Presented to
The Graduate Faculty of
The University of Southwestern Louisiana
In Partial Fulfillment of the
Requirements for the Degree
Master of Science

Spiros Triantafyllopoulos

July 1985

OVERVIEW

* * * INTRODUCTION

* * * DATA RETRIEVAL

* * * NATURAL LANGUAGE DATABASE FRONT ENDS

* * * GENERIC DESIGN OBJECTIVES

* * * AN INTRODUCTION TO KARL

* * * GENERIC OBJECTIVES REVISED

* * * DESIGN METHODOLOGY

* * * SPECIFIC NLQS DESIGN OBJECTIVES

* * * KARL NL PROCESSING CAPABILITIES

* * * OVERVIEW OF THE QUERY PROCESSING CYCLE

OVERVIEW (CONT'D)

*** DATA STRUCTURES

*** LEXICAL AND GRAMMAR ANALYSIS

*** SYNTACTIC ANALYSIS

*** SEMANTIC ANALYSIS

*** FORMAL QUERY GENERATION
AND EVALUATION

*** SYSTEM INTERCONNECTIONS

*** ANNOTATED EXAMPLES

*** EVALUATION OF OBJECTIVES

*** FUNCTIONAL EVALUATION

*** CURRENT STATUS AND FUTURE WORK

*** CONCLUSIONS

INTRODUCTION

*** APPLICABILITY OF COMPUTERS IN ALMOST EVERY HUMAN ACTIVITY

*** MORE APPLICATIONS ARE DEVELOPING

*** MORE NON-EXPERTS NEED ACCESS TO COMPUTERS

*** LACK OF COMPUTER LITERACY AMONG MANY CURRENT CASUAL USERS

*** MOST USERS EXPECT COMPUTERS WILL BE THE "SOLUTION TO ALL PROBLEMS"

*** FREQUENT USER DISSATISFACTION RESULTS

*** DEFINITE NEED FOR IMPROVED HUMAN-SYSTEM COMMUNICATIONS PROCEDURES

DATA RETRIEVAL

- *** THE INFORMATION AGE IS A REALITY
- *** WIDE VARIETY OF AVAILABLE TECHNOLOGIES AND SYSTEMS
- *** EARLY DATA RETRIEVAL TECHNIQUES
  - *** SIMPLE FILE-BASED SYSTEMS
  - *** LARGE APPLICATION PROGRAMS
  - *** LACK OF MODERN CAPABILITIES (I.E., SHARING, INTEGRITY)
- *** FILE MANAGEMENT SYSTEMS
  - *** IMPROVED PERFORMANCE
  - *** SOME CAPABILITIES FOR
    - *** SHARING
    - *** SECURITY
    - *** INTEGRITY
  - *** STILL, PROGRAMMING WAS NECESSARY

DATA RETRIEVAL (CONT'D)

*** DATA BASE MANAGEMENT SYSTEMS

*** SUPERIOR TO FILE MGMT. SYSTEMS

*** DIFFERENT TYPES

*** RELATIONAL

*** HIERARCHICAL

*** NETWORK

*** PROVIDE LANGUAGES FOR:

*** DATA DEFINITION/ORGANIZATION

*** DATA MANIPULATION/RETRIEVAL

*** CAPABILITIES FOR

*** SECURITY

*** DATA INDEPENDENCE

*** DATA REORGANIZATION

*** SHARING

DATA RETRIEVAL (CONT'D)

*** ACCESSING A DATABASE

*** INTERACTIVELY

*** BATCH MODE

*** THRU APPLICATION PROGRAMS

*** INTERACTIVE MODE MOST FREQUENT WITH CASUAL USERS

*** NO NEED FOR PROGRAMMING

*** MORE CONVENIENT

*** STILL REQUIRES FORMAL TRAINING

*** THERE IS A NEED FOR MORE EFFICIENT RETRIEVAL LANGUAGES

*** USER-ORIENTED LANGUAGES MOST APPEALING

DATA RETRIEVAL (CONT'D)

*** NATURAL LANGUAGE DATABASE QUERY SYSTEMS

    *** NON-PROCEDURAL LANGUAGES

    *** NO FORMAL SYNTAX OR SEMANTICS
    (SYSTEM LIMITATIONS MAY EXIST)

    *** REDUCED QUERY SIZES

*** CONSIDERING CASUAL USERS:

    *** MANY USERS LACK TIME OR DESIRE
    FOR FORMAL TRAINING

    *** USERS LACK SYSTEM KNOWLEDGE

    *** SYSTEM LACKS USER KNOWLEDGE

    *** RESULTS IN "KNOWLEDGE GAP"

NL DATABASE FRONT ENDS

*** RATIONALE FOR NATURAL LANGUAGE DATABASE QUERY SYSTEMS:

INCREASED USER EFFICIENCY THROUGH IMPROVED COMMUNICATIONS BETWEEN USER AND SYSTEM

*** NL QUERIES SIMPLER THAN ANY OTHER RETRIEVAL ALTERNATIVE (FORMAL QUERIES, PROGRAMS, ETC.)

*** EXAMPLE:

FORMAL QUERY:

```
RANGE OF E IS EMPLOYEE
SELECT (SALARY, NAME)
WHERE (SALARY > 18,000 & SEX = "MALE")
PRINT E
```

NL QUERY

PLEASE PRINT THE NAMES AND SALARIES OF ALL MEN THAT EARN MORE THAN $18,000

FORMAL VERSUS NATURAL QUERY

NL DATABASE FRONT ENDS (CONT'D)

*** MAJOR ADVANTAGES

- *** INCREASED HUMAN PRODUCTIVITY
- *** INCREASED SYSTEM PRODUCTIVITY (FEWER ERRORS AND RE-TRIES)
- *** REDUCED USER FRUSTRATION
- *** VIRTUAL ELIMINATION OF A TRAINING PERIOD
- *** CUSTOMIZED CAPABILITIES CAN BE PROVIDED
- *** IMPROVED HANDLING OF "NATURAL" LANGUAGE CONCEPTS (THESAURUS, SYNONYMS, ETC)
- *** POSSIBLE INTEGRATION INTO A TOTAL NL FRONT END ENVIRONMENT

NL DATABASE FRONT ENDS (CONT'D)

*** PROBLEMS WITH NL IMPLEMENTATIONS ON EXISTING SYSTEMS

- *** LONG DEVELOPMENT TIMES
- *** RESTRICTED APPLICATION DOMAINS
- *** POOR PORTABILITY BETWEEN OPERATING SYSTEMS/TOOLS
- *** SOME SYSTEMS DO NOT SUPPORT PRODUCTION LEVEL DBMS'S
- *** EXTENSIVE RESOURCE UTILIZATION

*** STILL, EXISTING NLQS'S ARE IN HIGH DEMAND BY USERS AT ALL LEVELS

*** MANY PRODUCTION MODELS AVAILABLE

GENERIC DESIGN OBJECTIVES

- *** ADAPTABILITY TO NEW APPLICATIONS
  - *** SYSTEM MUST BE USABLE WITH NO CODE MODIFICATIONS
- *** PORTABILITY BETWEEN DATABASE SYSTEMS AND OPERATING SYSTEMS
- *** REDUCED COMPLEXITY
  - *** MODULAR, INDEPENDENT DESIGN
  - *** SIMPLE IMPLEMENTATION
- *** EFFICIENCY
  - *** OPTIMIZED DESIGN
  - *** OPTIMIZED IMPLEMENTATION

AN INTRODUCTION TO KARL

*** KARL IS A:

KNOWLEDGE

ASSISTED

RETRIEVAL

LANGUAGE

*** RESTRICTED NATURAL LANGUAGE
DATABASE QUERY SYSTEM

*** KNOWLEDGE-ASSISTED
(OTHER TECHNIQUES ALSO USED)

*** EXPERIMENTAL VEHICLE
FOR THE DESIGN AND IMPLEMENTATION
OF NATURAL LANGUAGE QUERY SYSTEMS

GENERIC DESIGN OBJECTIVES REVISED

*** ADAPTABILITY

- *** KNOWLEDGE BASE CAN BE REDEFINED TO USE WITH NEW APPLICATIONS
- *** LANGUAGE-RELATED KNOWLEDGE (LANGUAGE RULES ARE TYPICALLY INDEPENDENT OF APPLICATION)

*** PORTABILITY

- *** KARL IS IMPLEMENTED USING:
  - *** "C" PROGRAMMING LANGUAGE
  - *** UNIX 4.2 OPERATING SYSTEM
  - *** INGRES V7 DBMS
- *** NO SYSTEM-DEPENDENT CALLS
- *** GENERAL EMBEDDED QUERY STRUCTURE

GENERIC DESIGN OBJECTIVES REVISED (CONT'D)

*** REDUCED COMPLEXITY

- *** COMMON PROGRAMMING LANGUAGE USED
- *** HIGHLY MODULAR DESIGN
- *** PRECISE MODULE INTERFACES
- *** SINGLE-FUNCTION COMPONENTS

*** EFFICIENCY

- *** NO DYNAMIC MEMORY ALLOCATION
- *** SIMPLE, EFFICIENT ALGORITHMS
- *** USE OF A COMPILED LANGUAGE
- *** REDUCED SUBROUTINE CALLS
- *** FURTHER OPTIMIZATION POSSIBLE

DESIGN METHODOLOGY

*** DIVIDE-AND-CONQUER APPROACH

*** DIVIDES TASK OF NL PROCESSING INTO A SEQUENCE OF SUB-PROBLEMS

*** DEFINES PRECISE INTEGRATION

*** SOLVES INDIVIDUAL PROBLEMS

*** INTEGRATES INTO FUNCTIONAL SYSTEM

*** FUNCTIONAL DECOMPOSITION

*** EACH MODULE PERFORMS A SINGLE TASK

*** MODULE SIZE DEPENDS ON FUNCTION

*** USES SOFTWARE TOOLS WHERE POSSIBLE

DESIGN METHODOLOGY (CONT'D)

*** TOP-DOWN INTEGRATION IS USED

- *** CONVENIENCE OF UPDATES/IMPROVEMENTS
- *** EFFICIENT DESIGN
- *** ERRORS ISOLATED IN SINGLE MODULES

*** INTEGRATION PROCEDURE

- *** COMMON QUERY REPRESENTATION AMONG DIFFERENT MODULES
- *** EACH MODULE IS VIEWED AS A "BLACK BOX"
- *** SEQUENTIAL PROCESSING ORGANIZATION
- *** PROVISION IS MADE FOR ERROR SIGNALS

SPECIFIC NLQS OBJECTIVES

*** KNOWLEDGE STORAGE, RETRIEVAL, ACQUISITION AND UTILIZATION CAPABILITIES

*** GRAMMATICAL AND LEXICAL CONSTRUCTS HANDLING CAPABILITIES

*** SYNTACTIC HANDLING CAPABILITIES

*** SEMANTIC HANDLING CAPABILITIES

*** ELLIPTIC QUERY HANDLING AND GENERAL ERROR REPORTING CAPABILITIES

KARL NL PROCESSING CAPABILITIES

- *** KNOWLEDGE CAPABILITIES
  - *** KNOWLEDGE ACQUISITION
    - *** AT DEVELOPMENT TIME
    - *** AT SETUP TIME
    - *** DURING ACTUAL USE
  - *** KNOWLEDGE REPRESENTATION
    - *** FRAME-BASED DYNAMIC KNOWLEDGE
    - *** RULE-BASED STATIC KNOWLEDGE
  - *** KNOWLEDGE UTILIZATION
    - *** IN ALL ASPECTS OF QUERY PROCESSING
    - *** EMBEDDED IN MODULES

KARL NL PROCESSING CAPABILITIES (CONT'D)

*** GRAMMAR/LEXICAL ANALYSIS CAPABILITIES

- *** DETERMINES WORD TYPES
- *** PERFORMS QUERY "CLEAN-UP"
- *** GENERATES DATA STRUCTURES

*** SYNTACTIC VERIFICATION CAPABILITIES

- *** OPERATES ON SINGLE DATA STRUCTURE
- *** A VARIATION OF NETWORK GRAMMARS IS USED (RECURSIVE TRANSITION GRAMMARS)
- *** DIFFERENT RTN FAMILIES HANDLED
- *** APPLICATION-INDEPENDENT PROCEDURE IS USED
- *** CAPABLE OF RESOLVING AMBIGUITIES

KARL NL PROCESSING CAPABILITIES

* * * SEMANTIC VERIFICATION
    * * * LINGUISTIC SEMANTICS
        * * * NOUN/VERB PHRASES
        * * * ADJECTIVE HANDLING
        * * * ELLIPSIS/AMBIGUITY HANDLING
    * * * DB VERIFICATION
        * * * QUERY SEMANTICS
        * * * INTEGRITY CONSTRAINTS
* * * LEARNING CAPABILITIES
    * * * UPDATE APPLICATION KNOWLEDGE
    * * * PROVIDE CUSTOMIZED PROCESSING
* * * ELLIPSIS AND AMBIGUITY CAPABILITIES
    * * * MISSING TERMS
    * * * USER CAN SUPPLY MISSING PARTS

OVERVIEW OF THE QUERY PROCESSING CYCLE

*** LEXICAL/GRAMMAR ANALYSIS

- *** IDENTIFY TOKENS/TYPES
- *** REPLACE SYNONYMS/REMOVE NOISEWORDS
- *** GENERATE DATA STRUCTURES

*** SYNTACTIC ANALYSIS AND VERIFICATION

- *** SUBMIT TOKEN TYPE LIST TO VERIFIER
- *** RECEIVE PATTERN FAMILY IDENTIFIER OR ERROR CODE (IF ERROR)
- *** USE PATTERN IDENTIFIER FOR FURTHER QUERY PROCESSING

OVERVIEW OF THE QUERY PROCESSING CYCLE (CONT'D)

*** SEMANTIC VERIFICATION

- *** VERIFY LINGUISTIC SEMANTIC CORRECTNESS
- *** VERIFY DATABASE SEMANTIC CORRECTNESS
- *** RESOLVE AMBIGUITIES/ELLIPSES

*** FORMAL QUERY GENERATION

- *** TRANSFORM TOKEN AND IDENTIFIER LISTS INTO GENERIC QUERY FORMAT
- *** GENERATE HOST DBMS QUERY

*** FORMAL QUERY EVALUATION

- *** OPEN DATABASE
- *** EXECUTE QUERY
- *** CLOSE DATABASE

AN OVERVIEW OF THE QUERY PROCESSING CYCLE (CONT'D)

```
       Input
       Query
        | |
        \/
+-------------+
|   LEXICAL   |        Intelligent Dictionary
|  ANALYSIS   |        Grammar Knowledge
+-----++------+
      | |
      \/
+-------------+
| SYNTAX      |        Syntax Knowledge
| VERIFIER    |        Schema Knowledge
+-----++------+
      | |
      \/
+-------------+
| SEMANTIC    |        Semantic Knowledge
| VERIFIER    |        Schema Knowledge
+-----++------+
      | |
      \/
+-------------+
| FORMAL      |        Schema Knowledge
| QUERY       |        Formal Syntax Knowledge
| GENERATION  |        Formal Semantic Knowledge
+-----++------+
      | |
      \/
+-------------+
| FORMAL      |        Formal Syntax Knowledge
| QUERY       |        Formal Semantic Knowledge
| EVALUATION  |        DBMS Specific Knowledge
+-------------+
```

THE NL QUERY PROCESSING CYCLE

DATA STRUCTURES

*** NL QUERY: LINKED LISTS

*** TOKEN LIST

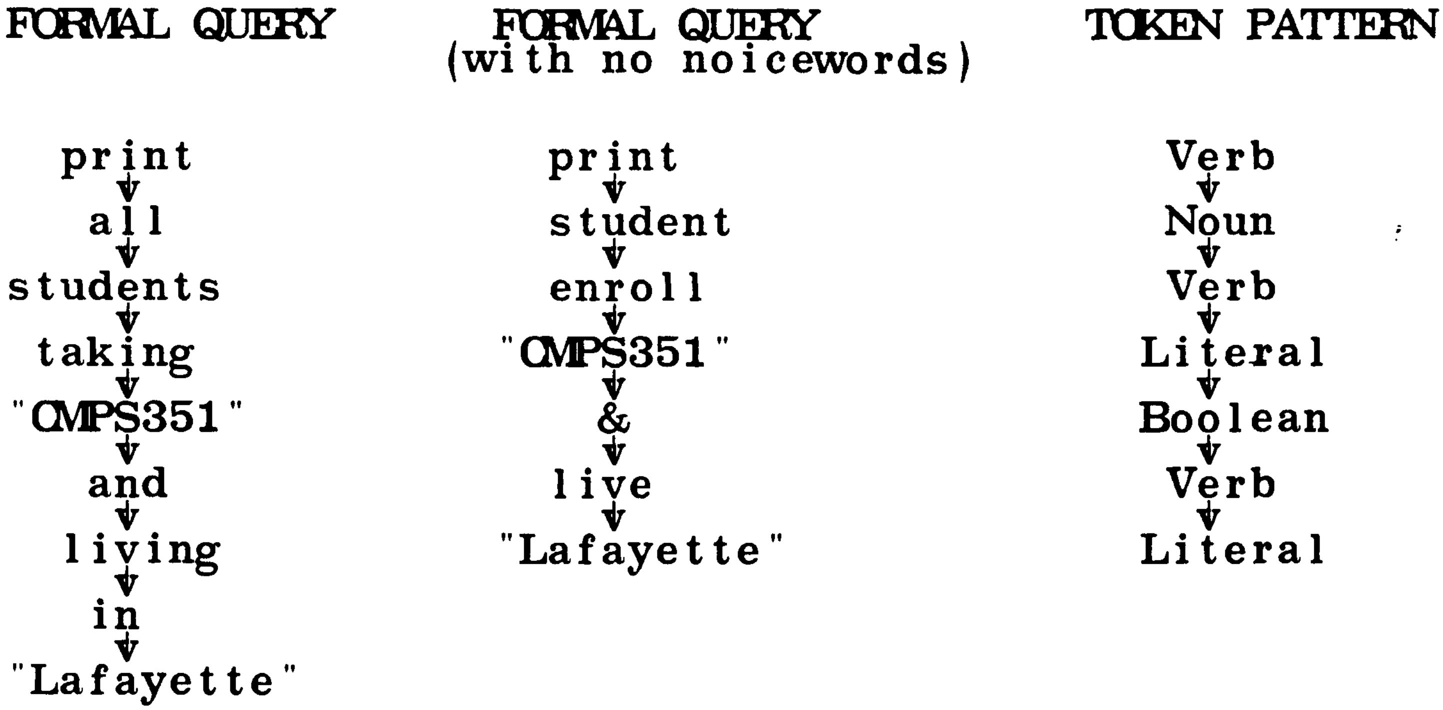
*** TOKEN IDENTIFIER LIST

```
+-----+-----------+        +------+-------+
| NO. |   token   |        | NO.  | type  |
+-----+-----------+        +------+-------+
      |                           |
      V                           V
+-----+-----------+        +------+-------+
| NO. |   token   |        | NO.  | type  |
+-----+-----------+        +------+-------+
      |                           |
      V                           V
  :   :   :   :   :          :    :   :   :
```

STRUCTURE OF NL QUERY STORAGE AREA

DATA STRUCTURES (CONT'D)

*** SAMPLE TOKENS AND TOKEN IDENTIFIERS:



FORMAL QUERY

print
↓
all
↓
students
↓
taking
↓
"CMPS351"
↓
and
↓
living
↓
in
↓
"Lafayette"

FORMAL QUERY
(with no noicewords)

print
↓
student
↓
enroll
↓
"CMPS351"
↓
&
↓
live
↓
"Lafayette"

TOKEN PATTERN

Verb
↓
Noun
↓
Verb
↓
Literal
↓
Boolean
↓
Verb
↓
Literal

*** LINKED LIST BASED IMPLEMENTATION

DATA STRUCTURES (CONT'D)

*** DICTIONARY
CONTAINS LIST OF ALL KNOWN WORDS AND TYPES

*** NOUN TABLE
CONTAINS LIST OF ALL KNOWN NOUNS, EITHER RELATION NAMES OR ATTRIBUTES

*** SYNONYMS TABLE
CONTAINS SYNONYMS AND EQUIVALENT TERMS

*** VERBS TABLE
CONTAINS VERBS AND RELATED NOUNS

*** ADJECTIVES TABLE
CONTAINS ADJECTIVES AND ASSOCIATED PROPERTIES ASSIGNED TO NOUNS

*** MULTIPLE SEQUENCE PATTERNS TABLE
CONTAINS NOUN SEQUENCES MAPPED TO SINGLE NOUNS IN THE KNOWLEDGE BASE

DATA STRUCTURES (CONT'D)

Noun Frame

Name	Type	Datatype	Max	Min	Pattern	Unit

Synonyms Representation

term	stands for

Verbs Representation

verb	subject	object

Adjective Representation

Adjective	Noun	Implied_property

Dictionary Representation

Word	Word_type

Multiword Representation

Term	Pattern_id	Rank

DYNAMIC KNOWLEDGE REPRESENTATION SCHEMA

LEXICAL AND GRAMMAR ANALYSIS

*** LEXICAL ANALYSIS

- *** IDENTIFY TOKENS
- *** ATTACH TOKEN IDENTIFIERS
- *** GRAMMAR TRANSFORMATIONS MAY BE NEEDED
- *** REPLACE SYNONYMS/REMOVE NOISEWORDS

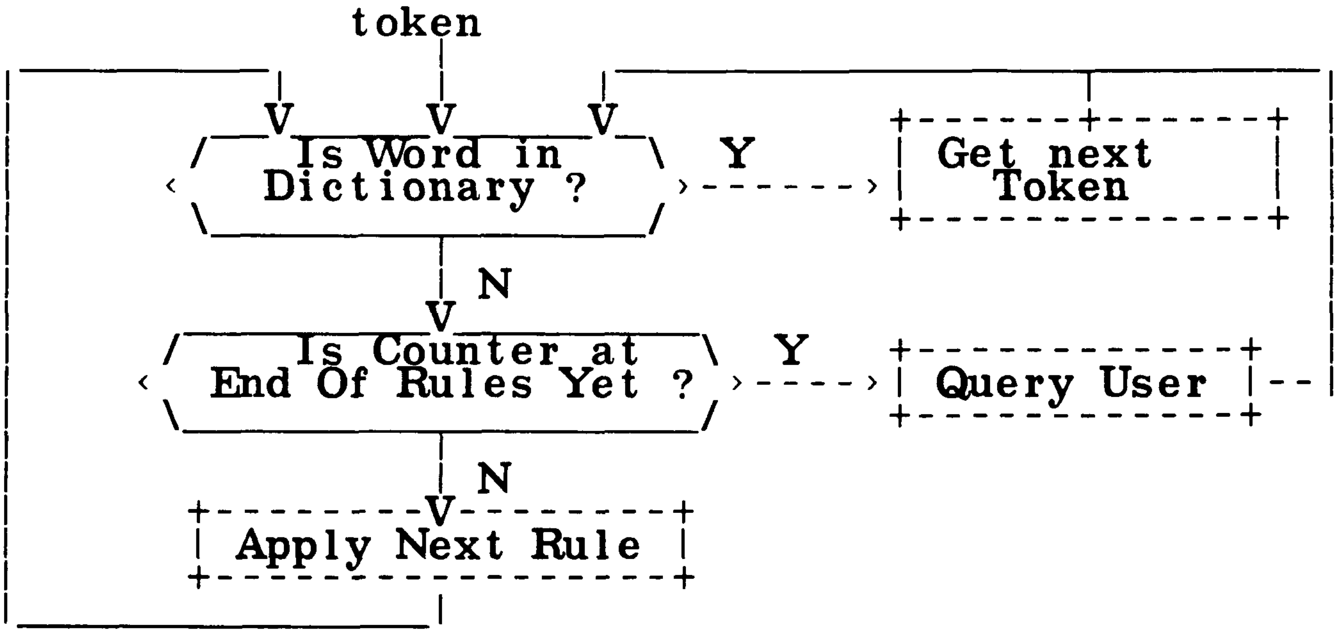
```
+--------------------+
| Read NL Query      |
+--------------------+
          |
+---------v----------+
| Replace Multiple   |
| Sequence Patterns  |
+--------------------+
          |
+---------v----------+
| Generate Tokens    |
+--------------------+
          |
+---------v----------+
| Replace Synonyms   |
| and Noicewords     |
+--------------------+
```

LEXICAL ANALYSIS OF INPUT NL QUERY

LEXICAL AND GRAMMAR ANALYSIS (CONT'D)

*** GRAMMAR ANALYSIS

- *** IF WORD IS KNOWN, THEN PROCEED
- *** USE RULES TO DETERMINE WORD TYPE
- *** QUERY USER IF UNKNOWN
- *** RULES ENCODED AS "C" FUNCTIONS



GRAMMAR ANALYSIS OF INPUT QUERY

SYNTACTIC ANALYSIS

*** VERIFIES CORRECTNESS OF NL QUERY BASED ON SYNTACTIC CRITERIA

*** MEANING OF ENTITIES NOT CONSIDERED

*** NETWORK-BASED GRAMMAR

*** TOKEN TYPES CURRENTLY SUPPORTED:

- *** NOUNS (N)
- *** ADJECTIVES (A)
- *** BOOLEAN OPERATORS (B)
- *** RELATIONAL OPERATORS (R)
- *** SYNONYMS (S)
- *** VERBS (V)
- *** LITERALS (L)

SYNTACTIC ANALYSIS (CONT'D)

- *** TOKEN SEQUENCES (PATTERNS)
- *** VERIFY RELATIVE POSITION OF TOKENS
- *** DIFFERENT PATTERN FAMILIES REPRESENTED
- *** EXAMPLE:

V (NB?)+ (VLB?)+	print names of students that live in "Dallas"
V (NB?)+ (NR+LB?)+	print names of faculty with salary of more than 24,000
V (AN)+	print the good students
V (VLB?)	who is working in "Dallas"? ("who" is replaced by "retrieve name")

(a)	repetitions of construct "a"
a+	one or more occurences of construct "a"
a?	construct "a" is optional
a*	zero or more occurences of construct "a"

SAMPLE PATTERNS AND QUERIES

SYNTACTIC ANALYSIS (CONT'D)

*** IMPLEMENTS RTN VERIFIER USING A FINITE STATE AUTOMATON REGULAR EXPRESSION RECOGNIZER

*** ACCEPT/REJECT STATES ONLY

*** 11 PATTERN FAMILIES SUPPORTED

*** IF NO PATTERN MATCHES, QUERY IS REJECTED

*** FINITE STATE AUTOMATON IMPLEMENTED THROUGH THE "LEX" LEXICAL ANALYZER GENERATOR SOFTWARE TOOL

*** "LEX" ACCEPTS FINITE STATE AUTOMATA SPECIFICATIONS AND GENERATES SOURCE CODE FOR REGULAR EXPRESSION VERIFIERS BASED ON THE SPECIFICATIONS

SYNTACTIC ANALYSIS (CONT'D)

*** "LEX" DESCRIPTION FOR SAMPLE RECOGNIZER:

```
[A-Za-z][A-Za-z0-9_]*        { return (IS_VARIABLE); }
-?[0-9]+                     { return (IS_INTEGER) ; }
-?[0-9\.]+                   { return (IS_FLOATING); }
"+-*/%"                      { return (IS_OPERATOR); }
```

*** "LEX" CONSTRUCTS

A-Z	matches single character uppercase
a-z	matches single character lowercase
0-9	matches single digit
[...]	groups sub-patterns
.	any character
*	zero or more times repetition
+	one or more times repetition
$	indicates end of line
?	optional element

SAMPLE LEX RECOGNIZER AND LEX CONSTRUCTS

SEMANTIC ANALYSIS

*** LINGUISTIC ANALYSIS

- *** NOUN PHRASE VERIFICATION
- *** VERB PHRASE VERIFICATION
- *** AMBIGUITY RESOLVING
- *** ELLIPSIS/PLETHORA HANDLING

*** PROCESS FLOW DIAGRAM:

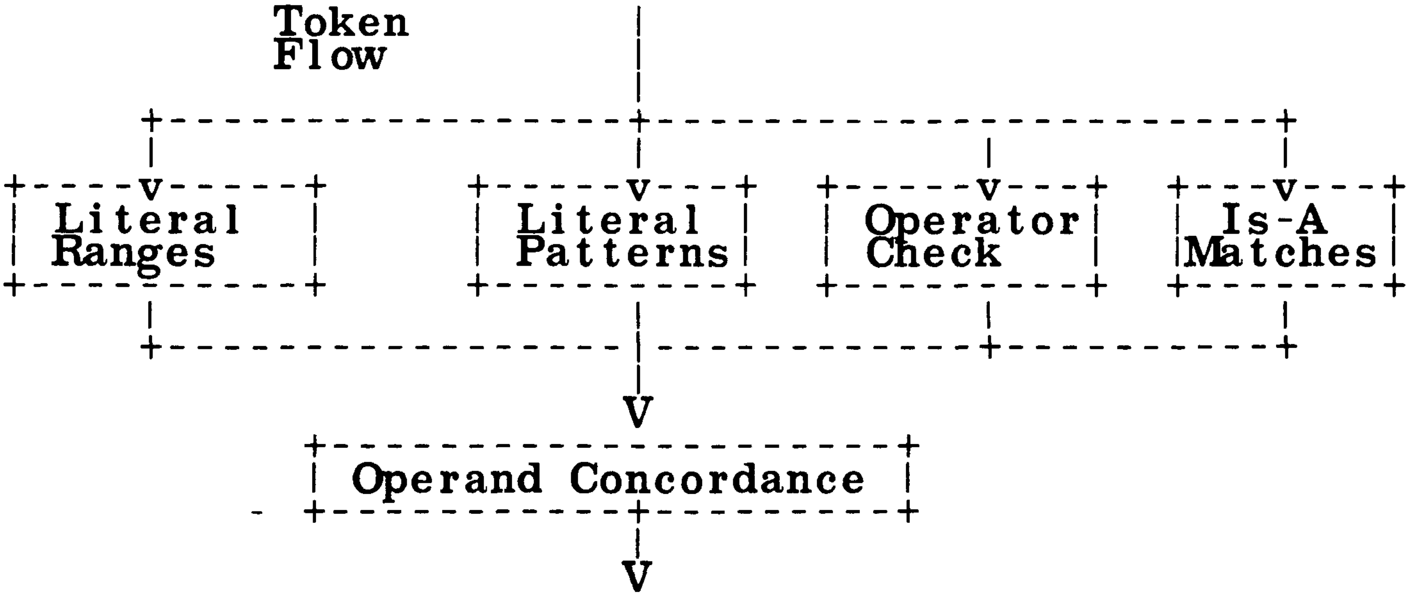
```
              Token           |
              Flow            |
                              |
       +----------------------+----------------------+
       |                      |                      |
  +----v------+         +-----v----+          +------v-----+
  | Ellipsis  |         | Plethora |          | Ambiguity  |
  +-----------+         +----------+          +------------+
       |                      |                      |
       +----------------------|----------------------+
                              |
                              V
                  +-----------------------+
                  | Verb Phrase Proc.     |
                  +-----------+-----------+
                              |
                  +-----------v-----------+
                  | Noun Phrase Proc.     |
                  +-----------------------+
                              |
                              v
```

SEMANTIC ANALYSIS (CONT'D)

*** DB VERIFICATION

- *** LITERAL RANGES
- *** LITERAL PATTERNS
- *** OPERATORS
- *** OTHER INTEGRITY CONSTRAINTS
- *** IS-A MATCHES (RELATIONSHIP MEMBERSHIP)

*** PROCESS FLOW DIAGRAM:



Token Flow

Literal Ranges

Literal Patterns

Operator Check

Is-A Matches

Operand Concordance

SEMANTIC ANALYSIS (CONT'D)

*** IMPLEMENTED THROUGH "C" FUNCTIONS

*** USES DYNAMIC KNOWLEDGE

*** BOTH RULE- AND FRAME- BASED

*** SAMPLE RULES:

```
IF   TOKEN(N) IS ADJECTIVE
THEN TOKEN(N + 1) MUST BE NOUN AND
     NOUN AND ADJECTIVE MUST AGREE
     AND HAVE ENTRY IN THE KB-ADJ.
ELSE ERROR = NO-NOUN-ADJ-AGREEMENT.

IF   TOKEN(N) IS VERB
THEN TOKEN(N-K), TOKEN(N+K) ARE NOUNS
     AND MUST AGREE WITH THE DEFINI-
     TION OF THE VERB IN THE KB-VERB.
ELSE ERROR = NO-VERB-NOUN-AGREEMENT.

IF   TOKEN(N) IS LITERAL
THEN TOKEN(N-K) IS THE NOUN ENTITY
     SO VERIFY THAT LITERAL RANGE
     IS ACCEPTABLE
ELSE ERROR = LIT-OUT-OF-RANGE.
```

FORMAL QUERY GENERATION AND EVALUATION

*** RELATIVELY SIMPLE TASK AS NL QUERY IS BEING "FORMALIZED" THROUGHOUT THE PROCESS CYCLE

*** DETERMINES DOMAINS/RANGES OF ATTRIBUTES

*** DETERMINE TYPE OF OPERATION REQUESTED (COUNT, EXIST, RETRIEVE, ETC.)

*** SELECT ATTRIBUTES TO BE RETRIEVED

*** STRUCTURE THE CONDITIONALS LIST TO CONFORM WITH "SELECT-FROM-WHERE" GENERIC QUERY FORMAT

*** CREATE GENERIC "SELECT-FROM-WHERE" QUERY AND DISPLAY IT TO THE USER

FORMAL QUERY GENERATION AND EVALUATION (CONT'D)

*** VERIFY GENERIC QUERY FOR CORRECTNESS (I.E., BOOLEAN OPERATORS MAY BE MISSING)

*** GENERATE HOST DBMS-SPECIFIC FORMAL QUERY

*** EXECUTE HOST DBMS-SPECIFIC QUERY

*** DISPLAY RESULTS TO THE USER

*** GENERIC AND INGRES QUERY FORMATS:

"Blank" Format:

```
    SELECT   ‹attribute_list›
    FROM     ‹domain›
   WHERE     ‹condition_list›
```

QUEL Format:

```
    RANGE OF ‹abbrev_name› IS ‹domain›
    RETRIEVE ‹dot_attr_list›
   WHERE     ‹dot_conditional_list›
```

(dot is the attribute domain prefix indicator)

SYSTEM INTERCONNECTIONS

*** INTERNAL MODULE CONNECTIONS:

```
                              Input NL Query
                                    |
                         +---------v---------+
                         | Lexical and       |
                         | Grammar Analysis  |
                         +-------------------+
                                    |
                         +---------v---------+
                         | Syntax Analysis   |
                         | and Verification  |
                         +-------------------+
+-----------+                       |                         +--------
| Knowledge |                       |                         | Error
| Base Mgmt +            +---------v---------+                + Handl-
| System    |            | Semantic Analysis |                | ing
+-----------+            | and Verification  |                +--------
                         +-------------------+
                                    |
                         +---------v---------+
                         | Query Generation  |
                         +-------------------+
                                    |
                         +---------v---------+
                         | Query Evaluation  |
                         +-------------------+
```

SYSTEM INTERCONNECTIONS (CONT'D)

*** EXTERNAL SYSTEM CONNECTIONS

```
+-----------------------------------------------------------------
| UNIX |
|------
|                                                  +--------+
|                                                  | System |
|       +---------------------------------+        | Calls  |
|       |                                 |        +---+----+
|       | The KARL System                 |            |
|       |                                 +------------+
|       +------+-------------------++----+
|              |                   ||
|     low      |                   || high
|    level     |                   || level
|              |                   ||
|       +------+-------------------++----+
|       |                                 |
|       |   INGRES Relational DBMS        |
|       |                                 |
|       +--+------------+------------+---+
|          |            |            |
|          |            |            +----------+
|          |            |                       |
|   :::::::::::   :::::::::::         XXXXXXXXXXXXX
|   :: data  ::   :: data  ::         X Knowledge X
|   :: base  ::   :: base  ::         X   Base    X
|   :::::::::::   :::::::::::         XXXXXXXXXXXXX
+-----------------------------------------------------------------
```

(All Data Paths Bi-Directional)

ANNOTATED EXAMPLES

QUERY 1:

show the students enrolled in "CMPS351" or "CMPS360"

```
LEXICAL ANALYSIS: show student enroll "CMPS351"
                               or "CMPS360"

     (ellipsis): show student enroll "CMPS351" or
                               enroll "CMPS360"

PATTERN:  Verb (Noun Bool?) (Verb Literal Bool?)*

SYNTACTIC ANALYSIS: Pattern Accepted, Pattern_No = 8

SEMANTIC ANALYSIS: enroll (student, course)
                   course PATTERN = "XXXX9999"
                   course Number  = 360 < 699

                   course Number  = 351 < 699

BLANK QUERY:       SELECT all  /* default */

                   FROM   student

                   WHERE  (course = "CMPS351"  |
                           course = "CMPS360"  )
```

QUERY PROCESSED CORRECTLY

ANNOTATED EXAMPLES (CONT'D)

```
QUERY 2:

who is "000-4076-65"

LEXICAL ANALYSIS: retrieve name "000-4076-65"
(severe ellipsis): retrieve name "000-4076-65"

PATTERN: Verb ( Noun Rel_op? Literal Bool? )+

SYNTACTIC ANALYSIS: Pattern Accepted, Pattern_No = 4

SEMANTIC ANALYSIS: Pattern "999-9999-999" matches ssn
          REFORMS: show student ssn "000-4076-65"
                   ssn  PATTERN = "999-9999-99"

BLANK QUERY:        SELECT name
                    FROM   student
                    WHERE  (ssn = "000-4076-65")

QUERY PROCESSED CORRECTLY
```

ANNOTATED EXAMPLES (CONT'D)

QUERY 3:

print names and addresses of all the rich faculty

LEXICAL ANALYSIS: print name address rich faculty

PATTERN: Verb (Noun Bool?)+ (Adjective Noun)+

SYNTACTIC ANALYSIS: Pattern Accepted, Pattern_No = 12

```
SEMANTIC ANALYSIS: name belongs to faculty
                   address belongs to faculty
                   rich := salary › 40,000
          REFORMS: print name address faculty
                   salary › 40,000
                   salary range acceptable
```

BLANK QUERY

```
SELECT name, address
FROM   faculty
WHERE  salary › 40000
```

QUERY ACCEPTED

ANNOTATED EXAMPLES (CONT'D)

QUERY 4:

show students who live and work in "Lafayette"

LEXICAL ANALYSIS: show student live & work "Lafayette

PATTERN MATCHED: NONE (although sentence is correct)

SYNTACTIC ANALYSIS: Failed. Program could not parse input sentence (No double verb pattern supported)

QUERY REJECTED

ANNOTATED EXAMPLES (CONT'D)

QUERY 5:
show the rich students

LEXICAL ANALYSIS: show rich student

PATTERN MATCHED: Verb (Noun Relop Literal Bool?)+
(severe ellipsis, matches after replacing "rich")

SYNTACTIC ANALYSIS: Pattern valid. Pattern No: 4

SEMANTIC ANALYSIS: rich student: error.
Attribute "salary" not associated
with relation "student"

QUERY REJECTED

ANNOTATED EXAMPLES (CONT'D)

QUERY 6:

show the students enrolled in "CMPS999"

LEXICAL ANALYSIS: show student enroll "CMPS999"

PATTERN MATCHED: Verb (Verb Literal Bool?)+

SYNTACTIC ANALYSIS: Pattern valid. Pattern No:_11

```
SEMANTIC ANALYSIS: enroll (student, class) OK
                   class pattern OK
                   class number out of range
                   class number > 699
```

QUERY REJECTED

EVALUATION OF OBJECTIVES

* * * DETERMINE IF GENERIC AND SPECIFIC OBJECTIVES HAVE BEEN MET WITH THE PROPOSED DESIGN

* * * GENERIC OBJECTIVES:

- * * * ADAPTABILITY
- * * * PORTABILITY
- * * * REDUCED COMPLEXITY
- * * * EFFICIENCY

* * * GENERIC OBJECTIVES HAVE BEEN MET THROUGH METHODOLOGY PRESENTED

EVALUATION OF OBJECTIVES (CONT'D)

*** SPECIFIC DESIGN OBJECTIVES

  *** KNOWLEDGE STORAGE, RETRIEVAL, ACQUISITION AND UTILIZATION CAPABILITIES

  *** GRAMMATICAL AND LEXICAL CONSTRUCTS HANDLING CAPABILITIES

  *** SYNTACTIC HANDLING CAPABILITIES

  *** SEMANTIC HANDLING CAPABILITIES

  *** ELLIPTIC QUERY HANDLING AND GENERAL ERROR REPORTING CAPABILITIES

*** SPECIFIC DESIGN OBJECTIVES HAVE ALSO BEEN MET THROUGH FOLLOWING THE GUIDELINES SET BY THE GENERIC DESIGN CRITERIA AND THE DESIGN METHODOLOGY PRESENTED

*** KARL 1.00 CAPABLE OF PROCESSING 60-65% OF QUERIE SUBMITTED (ADJUSTED FOR TYPING AND SPELLING ERRORS).

FUNCTIONAL EVALUATION

	CRITERION	KARL
1.	Be able to access multiple databases (i.e., retargetable within applications)	Y
2.	Answer questions asked directly (i.e., Who)	Y
3.	Handle multiple files and relationships	Y
4.	Handle simple pronoun references	N a
5.	Be able to handle ellipsis	Y
6.	Provide report generating facilities for the retrieved data (i.e., formats, graphs, etc)	N
7.	Be able to extend the linguistic knowledge of the system during program execution	Y
8.	Handle null cases, indicating the condition(s) that failed	N b
9.	Restate in English the user's query	Y c
10.	Handle spelling and typing errors	N
11.	Provide special functions for improvement the database capabilities	N b
12.	Provide semantic constraints in the dialogue between the human and the machine, and handle errors such as plethora and ambiguity	Y

(a) Item has been considered as future extension

(b) Item not in the original design considerations

(c) The program restates the semi-formally

CURRENT STATUS AND FUTURE WORK

*** CURRENT LIMITATIONS:

- *** NESTED QUERIES
- *** SPELLING CORRECTION
- *** NULL QUERY HANDLING
- *** PRONOUN REFERENCES

*** DYNAMIC KNOWLEDGE BASE STATUS:

255 TOTAL KNOWN WORDS
8 VERBS
7 ADJECTIVES
20 FRAMES
27 MULTIPLE SEQUENCES
24 NOUNS
45 SYNONYMS

*** CURRENT APPLICATION: UNIVERSITY DATABASE

CURRENT STATUS AND FUTURE WORK (CONT'D)

*** FUTURE RESEARCH TOPICS:

- *** NESTED QUERY HANDLING
- *** PRONOUN REFERENCES
- *** SPELLING CORRECTION
- *** NULL QUERY HANDLING
- *** INTERFACE WITH OTHER SYSTEMS (I.E., COMMON COMMAND LANGUAGE IS&R FRONT END, OFFICE AUTOMATION SYSTEMS, OR OTHERS)
- *** QUERY OPTIMIZATION

CONCLUSIONS

*** SIGNIFICANCE OF THE THESIS:

- *** AN ALTERNATE DESIGN METHODOLOGY FOR NLQS WAS INTRODUCED
- *** DESIGN CONSIDERATIONS AND METHODOLOGY APPLICABLE TO OTHER NL PROCESSING AREAS
- *** A FOUNDATION FOR FURTHER RESEARCH AND DEVELOPMENT WAS PRESENTED
- *** FURTHER RESEARCH TOPICS WERE IDENTIFIED
- *** SOLUTIONS WERE PROPOSED FOR SUCH TOPICS USING CURRENT PROTOTYPE AS A FOUNDATION

CONCLUSIONS (CONT'D)

*** NO NEED TO EMULATE OR SIMULATE NATURE

*** AN INVENTING RATHER THAN AN IMITATING APPROACH IS NEEDED

*** FUNCTIONAL EQUIVALENCE CAN OBTAIN SIMILAR RESULTS WITH SIMULATION/EMULATION, USING CONVENTIONAL TOOLS AND TECHNIQUES

*** FUNCTIONAL DECOMPOSITION CAN ASSIST IN REDUCING COMPLEX PROBLEMS INTO WORKABLE SIZE PROBLEMS

*** TECHNIQUES EXIST FOR SOLVING SMALLER SIZE PROBLEMS (COMPILER METHODS, SOFTWARE TOOLS, ARTIFICIAL INTELLIGENCE, ETC.)

CONCLUSIONS (CONT'D)

*** A NLQS CAN PROVIDE THE FOUNDATION FOR OTHER NL-BASED SOFTWARE SYSTEMS

*** DEFINED FUNCTIONALITY OF EACH COMPONENT WILL BE REQUIRED WITH NO INTERDEPENDENCIES

*** INTEGRATION TECHNIQUES WILL HAVE TO BE DEVELOPED TO MERGE ALL NL-BASED COMPONENTS INTO AN INTEGRATED ENVIRONMENT

*** THEN, THE "HUMAN COMPUTER" PROBLEM CAN BE ADDRESSED AND SOLUTIONS PRESENTED

4.23

1. Report No. IN-82

2. Government Accession No. 183568 ~~144378~~

3. Recipient's Catalog No.

4. Title and Subtitle 571

USL/NGT-19-010-900: KARL: A KNOWLEDGE-ASSISTED RETRIEVAL LANGUAGE: PRESENTATION VISUALS

5. Report Date DATE

October 31, 1985 JUL 2011

6. Performing Organization Code

7. Author(s)

SPIROS TRIANTAFYLLOPOULOS

8. Performing Organization Report No.

9. Performing Organization Name and Address

University of Southwestern Louisiana
The Center for Advanced Computer Studies
P.O. Box 44330
Lafayette, LA 70504-4330

10. Work Unit No.

11. Contract or Grant No.

NGT-19-010-900

12. Sponsoring Agency Name and Address

13. Type of Report and Period Covered

FINAL; 07/01/85 - 12/31/87

14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

This Working Paper Series entry represents a collection of presentation visuals associated with the companion report entitled "KARL: A Knowledge-Assisted Retrieval Language," USL/DBMS NASA/RECON Working Paper Series report number DBMS.NASA/RECON-21.

This report represents one of the 72 attachment reports to the University of Southwestern Louisiana's Final Report on NASA Grant NGT-19-010-900. Accordingly, appropriate care should be taken in using this report out of the context of the full Final Report.

17. Key Words (Suggested by Author(s))

KARL, Knowledge-Assisted Retrieval Language, Information Storage and Retrieval Systems

18. Distribution Statement

19. Security Classif. (of this report)

Unclassified

20. Security Classif. (of this page)

Unclassified

21. No. of Pages

54

22. Price*

*For sale by the National Technical Information Service, Springfield, Virginia 22161